smsqzine

Issue #6   November 2018

smsqzine

table of contents

Editorial

You'll see a trend in this issue of taking another look at Xchange. As I have gone back to Xchange / Quill for writing, I've had to come up with the tools and workflow to make this work. After an aborted attempt at using Quill for MS-DOS, I've gone back to Quill for the QL. Using both the MS-DOS and QL version of Quill, I was curious of the history of the Psion business program and tracked down the the history of the applications.

I like reading about how others go about using the different tools on our computers. You never know what someone has thought of something that did not strike you before and find out it is a really good idea. Hopefully my discussing the tools that I am using for writing might influence someone.

Bob Dyl

Bob Dyl was a key QL personality in the 90's in the US. He had gone from a fairly obscure Sinclair vendor to creating one the largest QL publication in the US. ......

Bob Dyl started the "English Micro Connection" in about 1984 or 1985. He imported Spectrum software and sold Spectrums. He was not a Spectrum importer, but he acted as an agent for UK vendors. His advertisements said that he would take payment, forward that to the UK vendor selling the Spectrum and the Spectrum would be shipped directly from the UK.

He might have been one of the only US vendors selling the Spectrum. Most of the US vendors were focused on the T/S 2068 and T/S 1000. A few US vendors would get Spectrum programs ported to the T/S 2068 and sell those. Other vendors would sell a Spectrum ROM-board that would fit in the cartridge slot of the T/S 2068. This would allow the T/S 2068 to run unmodified Spectrum software.

I saw reference to a possible NTSC version of the Spectrum that might have come from English Micro Connection, but if Bob was having the Spectrums sent from the UK, I doubt that they were converted to the US TV standard.

Bob also worked with Zebra Systems to get the Timex FDD disk drive system from Timex Portugal imported into the United States. There were two T/S 2068 disk systems, the Larkin and the Olinger, but the Timex FDD was a dual drive system designed to look like the T/S 2068.

At some point, Bob switched to the QL and started the "QL Connection", selling QL's. I found very little information about the "QL Connection" so I'm not sure if he imported QL hardware and/or resold US QL's from Sinclair or A+ Response.

In 1994, Bob was thinking about a QL Newsletter. There were existing user group and other Timex/Sinclair newsletters, but most covered all three computers without focusing on the QL. In about May 1994, Bob started the "International QL Report". Through his contacts in Europe, he was able to get information about new QL products.

He also helped organize some joints buys of QL hardware. This helped save on shipping, where the UK vendor would ship a quantity of product to a single address in the US.

The first year of IQLR the issues stayed fairly small. The first issue was 12 pages. The fourth issue was 20 pages. The first issue of the second volume jumped to 42 pages. Volume three started with 46 pages, but went to 52 pages with issue #3 and issue #4 jumping to 72 pages. The next two years most issues of IQLR were between 60 and 70 pages.

IQLR started getting a number of UK and German advertisers.   There was a few US QL vendors advertising, but the largest ads were from Europe. With more vendors and users having access to e-mail and credit cards, it was easier than ever to purchase products from Europe, either in pounds or marks.

In 1993, Bob Dyl started the "Miracle in Newport" QL shows. In 1993 and 1994 Bob organized a QL show in Newport, Rhode Island. He was able to get a number of European vendors to make the trip. These two shows started the annual QL show in the US. If I remember correctly, the 1995 QL show was held by the New England QL group near Boston. In 1996, the show moved to Oak Ridge, Tennessee. With out Bob Dyl starting the QL shows and showing that there is enough of a market to bring over the European vendors, the US QL scene would have been much different.

The last issue of IQLR was Volume 5, Number 6, March/April 1996. Bob had some health problems that brought his activity to a screeching halt and he quietly left the QL scene. With the loss of IQLR to the community, a few stepped up and created "QL Today".

ZX81 Development on the QL (part 2)

After playing with Basic in the last issue, time to turn to doing ZX81 development in assembly language. Gunther Strube write a Z80 cross-assembler that runs on the QL and MS-DOS called z80asm. The intention was for creating programs for the Z88, but the assembler can be used to create programs for the ZX81.

There is a z80asm that comes with Z88dk (the Z88 development kit), but it is a more recent version of the cross assembler. The version available for the QL is 1.01 from 1996. After using z80asm from z88dk, it took me awhile go get running with this early version of z80asm.

The cross assembler is available from Dilwyn's website as qlz80asm.zip. It comes with a Quill document that describes how the assembler is used. One key problem is a lack of examples, so it took some trail and error to get it running.

When I started learning ZX81 assembly, I found a number of header files that set up the first part of the .P file. It sets up the system variables that are in the first part of the .P file, then sets up the REM statement and then the BASIC line that calls the assembly.

I first started using these with TASM and that used one syntax for the labels and mnemonics. When I moved to Pasmo, that took another syntax for the labels and mnemonics. Z80asm under Linux uses a similar syntax to Pasmo. The problem is that the early version of z80asm used a different syntax. It was not well documented in the documentation, so it took a lot of trial and error to understand the new syntax.

The biggest changes in between the Pasmo syntax and the early z80asm are:

1. Labels start with a period.

In PASMO this is a valid label:

        loop:

In early z80am it as to start with a period:

        .loop

2. Most of the mnemonics work between Pasmo and early z80asm, but there is one difference:

Pasmo	z80asm
DEFB	DEFB
DEFW	DEFW
DEFS	DEFS
ORG	ORG
EQU	DEFC <-- Note

3. $ as an operator

With Pasmo, the $ is also an operator that gives the value of the position counter at the begin of the current sentence. Here is an example:

```
line0:
  DEFB $00,$01        ; line number
  DEFW line1-$-2      ; line length
```

With z80asm there is no $ operator, so the code has to be changes as follows:

```
line0:
  DEFB $00,$01                  ; line number
line01:
  DEFW line1-line01-2           ; line length
```

4. z80asm is case insensitive while Pasmo is case-sentative

VARS and vars are different with Pasmo.

```
VARS:        DEFW vars
DEST:        DEFW 0
E_LINE:      DEFW vars+1
```

With z80asm VARS and vars are the same, so I had to rename one of them.

```
VARS1:       DEFW vars
DEST:        DEFW 0
E_LINE:      DEFW vars+1
```

There is a -nocase option with Pasmo to make it case insensitive.

5. Special characters are not allowed in labels

```
_K  EQU  $30
```

```
DEFC .xK=$30
```

Using the Assembler

```
exec z80asm;"-b hello"
```

This will create a hello_bin file. Rename this to hello.p and then load it into Xtricator.

For those who still like to use QL hardware (the BBQL, Q40, Q60 or Q68) z80asm provides a tool to do ZX81 development on the QL.

PC Four to QL Xchange

I've been writing using Open Office for a number of years, but I've always had issues with the white background and the black letters. It's just too bright and harder on the eye then the old word processors that used a black background and white or green letters.

For a recent trip I decided to do my writing using Quill under DOS, part of the PC Four suite from Psion. After writing a number of pages, I had problems converting the document over to another format. The only thing I could do was to save the document as text using the Export command. Now I had to recombine the lines back into a paragraph.

I liked using PC Four with DosBox because I could get DosBox to go full screen. The difficulty in converting from PC Four to something else, made me move from DosBox and over to SMSQmulator and use the QL version of Xchange. It looks the same and almost goes full screen, but the main reason is that there are a number of tools to convert QL Quill documents to other formats like Rich Text Format (RTF), which most word processors support.

I started thinking more about the ability of conversion between PC Four file formats and QL Xchange formats. Looking at PC Four, the formats for all of the files (Quill, Abacus and Archive) are different

than the QL. It is not possible for either application to read files from the other version.

Looking at the commads on each application, Abacus has a Transfer option. This allows the transfer of files from PC Four to QL Abacus and back again. It retains all of the data and formula in the file.

QL Quill has Transfer option, but the PC Quill does not, so you can't use that to transfer documents. The only option is to export to text and import on other version.

Archive also does not have a Transfer option, but it is fairly simple to export a database into a comma separated variable (CSV) file and then import it into the other version of Archive. If that does not go too smoothly, once the database is in a text format, a little bit of programming can be used to get into the right format for importing.

The version of PC Four that I have does not come with Easel. I did find a copy of PC Xchange, but that version of Easel hangs. This Xchange also does not have Transfer for Quill or Archive, so it is basically the same as the PC Four versions.

So, I see no advantages of using the PC Four suite with DosBox. With either the PC or QL version, I still need an emulator, so why not just go with the QL version.

History of Psion Xchange

Every QLer knows about the 4 applications, written by Psion, that came with every QL. Quill, Abacus, Archive and Easel are probably the most used applications on the QL. Most QLer's know how the 4 programs were combined to create Xchange. This article will detail the history of the 4 Psion programs and Xchange.

When Sinclair was creating the QL they wanted a suite of business-type programs to run on it. This was not something new. The Osborne 1 was sold with Wordstar and SuperCalc. The Kaypro computers came with PrefectWriter, PerfectCalc, PerfectFiler, and PerfectSpeller. The idea was to increase the value of the computer by adding software that had a high retail value and make the computer turnkey to get the user up and running. Sinclair sent out a query to a number of computer software houses, asking for a proposal on an application suite.

Psion had a history of writing game software for Sinclair, starting with the ZX81 and then the Spectrum. As luck would have it, Psion was working on such a suite of programs that Sinclair was asking for. Psion had made the decision to diversify from games and get into the applications market.

Charles Davies, Psion's technical director said:

"Sinclair contacted a number of companies at the beginning of 1983. At the time they were extremely vague about the machine, just that it would 'probably be 16-bit'. They invited us to submit proposals for a suite of business software. Since we'd already given the matter a lot of thought, we were able to give Sinclair detailed proposals, which they accepted."

Psion had previously made a decision to develop their software on a DEC VAX minicomputer, then port the programs to the desired CPU, such as the Z80 or 6502. With almost no detailed information about the QL, Psion started development using the C language on the VAX hoping it would take minimal effort to port to the QL.

Charles Davies went on to say:

"The packages had to be easy to use for someone with no experience of or interest in computers. The whole emphasis was sitting down and using the packages from day one - without needing to refer to the documentation. We all know the 'if all else fails, read the manual' attitude; we tried to work with this attitude instead of against it. We've provided

substantial 'in-context' help facilities in all four packages. The prompt box at the top of the screen tells you exactly what you can do at any time and provides brief prompts. And the in-context help screen can be called for more detailed instructions."

"We were given a rack system in the summer of 1983. It had no microdrives, of course, and we still didn't know exactly what the keyboard would be like. But in all other respects the system was a QL."

"We didn't receive a copy of QDOS until a few weeks before the press launch in January. We had the choice of writing our own operating system to work with, or waiting for Sinclair to produce QDOS. Knowing it would be foolish to wait for Sinclair to produce QDOS, we developed our own operating system for development and testing, substituting QDOS when it arrived."

"The Sinclair hardware was delayed, so we used the time to improve the specifications of the software. There were a lot of different versions of both the hardware and Psion packages, and this caused problems with compatibility.   But the versions which have gone out to customers were fully working."

Three of the packages a very similar look and way of working, these being Quill, Easel and Abacus. They were easy to use and required limited references to the manual. Archive was different in that despite having a graphical look, to get most things done, some programming was required.

"With the three other packages our prime concern was ease of use. In some areas, we've sacrificed some of the more obscure features for the sake of simplicity. With the database, however, we found from talking with experienced users that they wanted was power. For this reason, we decided on a language driven system. This is inherently less friendly than a menu-driven, single-key system, but provides the sophistication that we know users want. The Archive programming language was designed to be as close to Basic as possible; if users know any programming language, they'll know Basic."

"The Archive programming language is much easier to use. It provides full prompting and in-context help at all times. You also get a lot of dedicated commands, like automatic sorting. There's a full editor, though you could of course use Quill if you prefer. Basic programmers will find very little difficulty with Archive, but we have to cater for inexperienced users too."

Martin Stamp developed Quill and provides some detail on how it was created:

"The main problem was space; we wanted to cram a lot of features into a small area. With the relatively slow speed of the Microdrive access, we couldn't rely too much on overlay files. The bit-mapped screen presented both problems and opportunities. The slow speed of writing to the screen meant that we couldn't subscribe to the 'when in doubt, redraw the screen' school of programming. We had to keep track of what was on the screen at any given time. But it has its compensations, enabling us to display underlining, superscripts and subscripts and soon."

"The emphasis throughout the word processor was on ease of use. We didn't want lots of complex features that nobody ever uses. You won't find a paragraph-delete function, for example. It would be too much power; people can go terribly wrong that way."

"Block definition is achieved using the cursor keys so that users can see what they are doing. Experienced computer users are quite happy with abstract concepts - they know that the block has been defined, even if they didn't see it happen. But when a beginner does something, he wants to see it happen. That way, he's sure."

Colly Meyers wrote Abacus and talks about its development:

"We feel that Abacus is much friendlier than most spreadsheets. There's none of this 'cell D45' business. If you want to refer to the cell containing the profits made in March, you simply specify 'Profits - March' and Abacus will find the cell where row marked 'Profits' crosses the column marked 'March'."

"As with the other software, Abacus is intended to be simple to use. Anyone can have the package up and running for straightforward applications within a matter of minutes. But the more complex and powerful features are there when you need them. And as with Archive, we've used Basic functions where ever possible since these - if anything - will be familiar to the users. So to find the length of a cell's contents, for example, you use LEN. We think that Ababus compares well with traditional spreadsheets in terms of speed and power, but is friendly enough for people to be tempted to use it to balance their bank account."

Martin Brown wrote Easel and talks about its development:

"Most business graphics packages are 'post-processing' programs. This is, they take data from, say, a database and turn it into a set form of graph or table - and that's it. We wanted to go for a genuinely interactive approach. With Easel, you can sit down in front of it, type in a few figures and instantly see those figures displayed as a bar-chart. If you want to turn the chart on it's side, you can. If you want a graph instead or as well, you can do that too."

"To start with, everything is set by defaults. You won't be asked a lot of questions first; you just get the default display. Later, you can change anything you don't like - colors, type of grid, form of chart or graph, and so on."

"You don't want to go to all the trouble of setting up a spreadsheet model, exporting the file to Easel, and loading Easel before you can product your graph. Easel has simple spreadsheet functions incorporated, enabling you to produce sine waves and so on."

The direct port from the Vax was version 1 of the suite. It had some issues with performance so version 2 was released in March 1985, which was programmed in machine code.

PC-Four

Since the development of the applications were done on the VAX and ported to the QL and based on their original decision to get into the business market, Psion ported the four applications to MS-DOS. The four stand-alone applications were called PC-Four.

ACT imported the Sirius 1 computer, known in the United States as the Victor 9000, into the UK for a number of years. The Victor 9000 was an MS-DOS computer that was "mostly" compatible with the IBM PC, but some applications written for the IBM PC would crash. It was not a complete clone of the IBM PC. ACT contracted with Psion to release PC-Four with the Sirius 1. ACT later developed the Apricot computer and it was also released with PC-Four.

Xchange

Psion also worked on a more advanced version of the 4 applications and created Xchange, which was a single application that incorporated all of the applications into a single executable. Xchange allowed for task swapping, meaning that a user can edit a document in Quill, and then switch out of Quill and fire up a version of Abacus. The user can switch back and forth between the two applications.

Xchange also had a Task Sequencing Language (TSL), which was a small language for automating Xchange commands, very much like a macro language.

Xchange replaced PC-Four on the Apricot and Sirius. Xchange was also distributed for the Touch Graduate IBM Compatible expansion device for the BBC Micro B. The Touch Graduate allowed the BBC Micro B to run MS-DOS programs. The Graduate was a PC in a box, but it used the BBC Micro's screen and keyboard via an expansion cable.

Stephen Morris wrote a book on Xchange and modified it for the two companies that were releasing Xchange; "Introducing Psion Xchange on the Sirius",

"Introducing Psion Xchange on the ACT Apricot" and the more generic "Introducing Psion Xchange on the IBM PC". These books were released in 1985.

In November 1986, Psion released PC-Four to the general public. The cost was 69 pounds (excluding
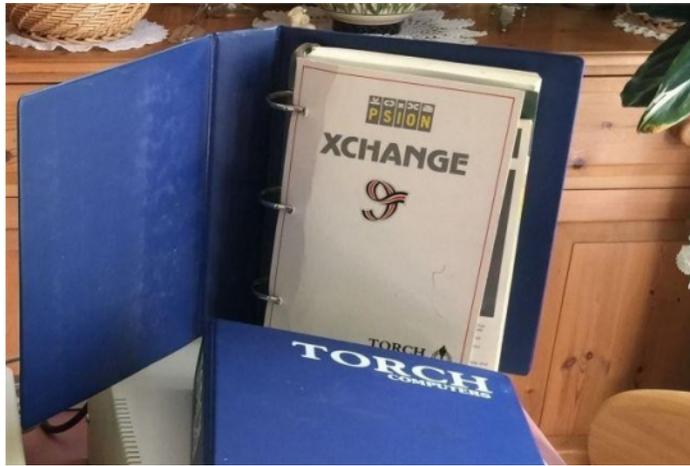
VAT). By summer of 1987, 10,000 copies had been sold. Xchange was also released to the general public. There was even a version that supported PC networks like Novel Netware, Microsoft MS-Net and the IBM Token Ring system. Xchange for MS-DOS was sold for 495 pounds.

In 1986, CST came out with the Thor computer. In 1987, CST and Dansoft released the Thor XVI computer. Dansoft wanted to do more than just have the 4 application programs. Dansoft contracted with Psion to create a version of Xchange for QDOS. Development was paid for by Dansoft and they had an exclusive agreement with Psion for the release of Xchange for the QL.

In 1987, Psion was still selling PC-Four and Xchange as an integrated business package, competing with other suites, like Lotus Symphony. JHP Training Limited started standardizing business computer training on the PC-Four suite. Softa and Co., a training and consultancy firm, offered a support contract on PC-Four.

In 1988, Psion started selling PC-Four and Xchange into the United States market. The Open University Computer Society (OUCS) in the UK recommended the use of PC-Four for its members. Also that year,

the Xchange Users Association was formed. The group was formed by Xchange "business users, dealers and distributors, VAR's, universities, hospitals and professional organizations." The group held their first conference in November of 1988. In August of 1988, Psion announced that PC-Four was available on 3.5" disks.

It is unclear when PC-Four and Xchange were no longer offered for sale, as this is not normally documented in the press. The Xchange Users Association was only mentioned in the PSION newsletter, so it is not documented when it stopped.

Quill to RTF

Now that I'm using Quill for writing, I need a way to convert Quill to a format that Open Office can use. Rich Text Format (RTF) is an older format that most word processors can read and write. There are a few programs on the QL that will convert a Quill document to an RTF document.

Looking on Dilwyn's website, I found three programs that will work:

Doc2RTF2 - Originally written by Mark Swift, it was updated by Thomas Kral. It is a SuperBASIC program that is compiled for speed. The original source code does come with the program.

QL2PC - A formerly commercial program from Geoff Wicks. This a Pointer Environment program that comes with a manual in PDF format.

Quill2RTF - Written by Pedro Reina. This C program comes with the source code.

Before starting my test, I created a Quill document with some test text, including Bold, Underline,

Superscript and Subscript. I was then going to compare the different programs against this test document.

I first tried Quill2RTF and found that the entire program is written in Spanish. The manual is also in Spanish. The program looks nice, but I don't know enough Spanish to get it working.

Doc2RTF2 is a simple program, asking just for the input file and the output file. The program was quick and RTF output file was created. I loaded the RTF file into Open Office. The program does a good job of keeping the look of the original document. The font used is Courier 10, a mono-spaced font. The three margins used by Quill are kept. For my documents, I am changing the default margins, but having the document default to Courier is not what I'm wanting. Bold and underline (even when combined) are kept in the RTF document. The Super- and Subscript were ignored.
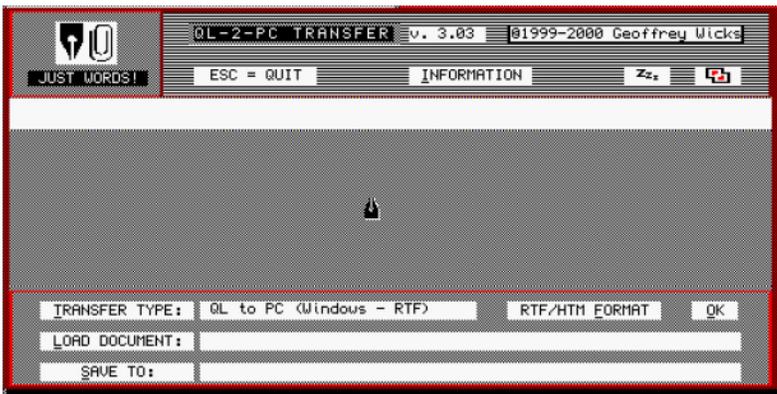
QL2PC is a pointer driven program. After it files up, I had to select the type of conversion I wanted. Using the normal PE entry fields, I entered the source document and the target document. I then told the program to convert. Looking at the finished RTF document, it looked better with Quill2PC. The document took the default font for the system (Times New Roman). Bold and Underline look fine. Superscript and subscript are supported and show up as lower and upper smaller fonts. The default margins were ignored to the document in Open Office looks more normal.

Most Expensive QL Software

In looking over some old QL paperwork that I have, I ran across a catalog from EMsoft, owned by Peter Hale of Boston. Peter wrote some Archive-based programs that he used and then sold to others. He was also good at finding others that had written QL software and trying to sell it.

One company that he re-sold for was Mechanical Design Analysis Consultants. They had written some software for mechanical engineering. One program was SECTPR for "analyzing the values of non-standard structural shapes that use modified or welded plate configurations." Another program was PEARWML which is a "statistical program to calculate Pearson III parameters.." This program was available for the QL, ZX81 and MS-DOS.

From the description, this sounds like some engineer wanted to automate that he was doing manually, got a hold of a ZX81 and started writing code. When the QL came out, he probably moved to the QL. I'm sure he bought the Sinclair line because they were cheaper that other computers.

The program that really caught my eye was CAM Master by MJM Software. The software is for controller a CNC milling machine. CNC stands for Computer Numerical Control. This is basically a computer controlled milling machine for metal. The device has a bit that grinds away at the metal slowly taking off metal until the desired shape is reached. These milling machines are expensive and by using a computer to control it, the machinist can set up the machine and then walk away while the milling takes place.

CAM Master is advertised as "read to run out of the box. No need for extensive training in CAD/CAM programming; indeed, with no prior computer experience, personnel will be productive before the end of the day."

The software requires a JSU ROM (so it was written in North America), 256K of memory and dual disk drives.

The price quoted in the catalog: $2,850

That is far more than the cost of the most expensive QL set up ever. I could have bought the QL at its highest price, bought the Super Gold Card with ED

```
Requires  a  JSU ROM  Sinclair  QL  computer  with  256K  memory
expansion and dual disk drives.  Available only on disk.

Base price for site license agreement:              $2850.00
Complete with computer/monitor/drives, etc.              Call

Demonstration disk with '_doc' file is available.  Cost is applicable to
the purchase of the site license.         (FLP $9.95 5.25" DSQD disk)
from MJM  Software
```

drives and still would not have cost this much.

There are two things to think about; one, how much time and effort did the programmer need to create the software. He also had to have a CNC milling machine to test it on. The second, how many people actually bought this software. I'm sure it is a fairly small market, but all you would need is a sale or two to make your money back.

Editors

With the recent release of the newest version of QED, I thought I would take a look at freeware editors. Since I spend a lot of my time writing, editors are near and dear to me.

I mostly wanted to see what kind of footprint each editor has. How light was the executable and how light was the installation. Most of the editors need few files, but some, like MicroEmacs, need more than the executable.

QED	- 10.7K
ED	- 21.5K
MasterSpy	- 38.9K
sedit	- 70.5K
QD	- 85.6K
The Editor	- 108K
Elvis	- 118.7K
MicroEmacs	- 208.3K

QED comes in as the smallest editor. Written entirely in assembly, it is a lean machine. ED, from Metacomo comes next (although it is still commercial). QED is a clone of ED.

MasterSpy is next. I've not used it so, I'm not sure if is has a large feature set. The odd editor, sedit, is next. This is one that I've never really used. Next is QD, the feature rich editor, also written in assembly. With fancier graphics than QD, it needs more space for them. Since QD has only recently become freeware, I have not used it.

The Editor, written in SuperBasic and then compiled, is next on the list. Compiled SuperBasic is not known of being small, but it is easier to code in for most of us. Elvis, the clone of the Unix editor, VI, is second to the largest. The source code is written to be very portable which gives it some size.

The largest is MicroEmacs. Also ported to the QL with extra code for the Pointer Environment. It is probably the most feature rich of the editors. The latest version has the ability to highlight the code from different programming languages. It has its own language for writing macros. It has the largest manual of the bunch. This used to be my go to editor, but I needed to use it often enough to remember all of the different commands. A lot of times I would use it instead of Quill, since it was quick and had a word wrap feature. MicroEmacs is available for other OS's, so I would also use it on MS-DOS.

Originally I was an ED user, as I got it as when I ordered the Metacomco Pascal compiler to use in college. Only more recently have I migrated to QED, which has the same command set as ED. Now, with the new features of QED, there is no reason to go back to ED. Being so small, I can put QED on all of the bootable QXL.WIN files that I have.
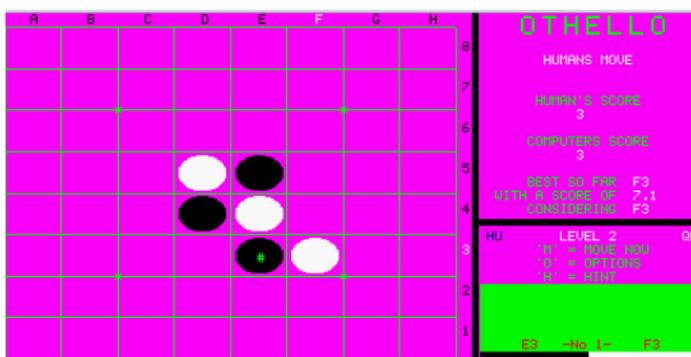
QL Othello

I'm more of a board game person that a video game person. One of the classic board games that I like is Othello, known generically as Reversi. There are a number of versions for the QL and I thought I would do a quick review of them. All of these versions are available from Dilwyn's Games page.
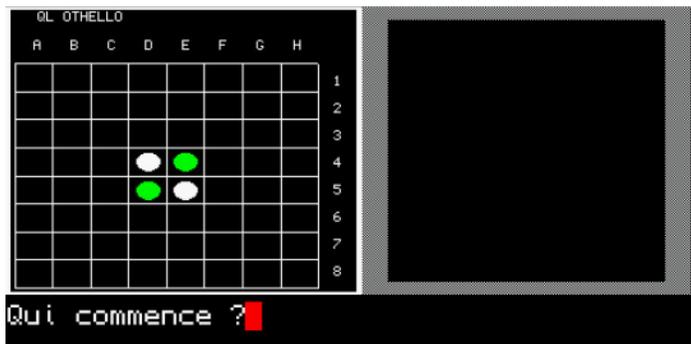

French version

The first game is othello2.zip. It is a French game, but although I could get it to run, but I don't know French, so I could not make it work.

The next is othello.zip, written by Kenneth Tait. It has an odd light purple background, it has a number of features, of which the best one is a hint option. Other than the odd color it plays a fairly decent game.


Kenneth Tait's version

Othello1.01.zip is a C version of Othello written by Richard Kettlewell. It will not run under SMSQ/E, but is fine running with QDOS (tested with uQLx). The game has a nice look, but the program played a
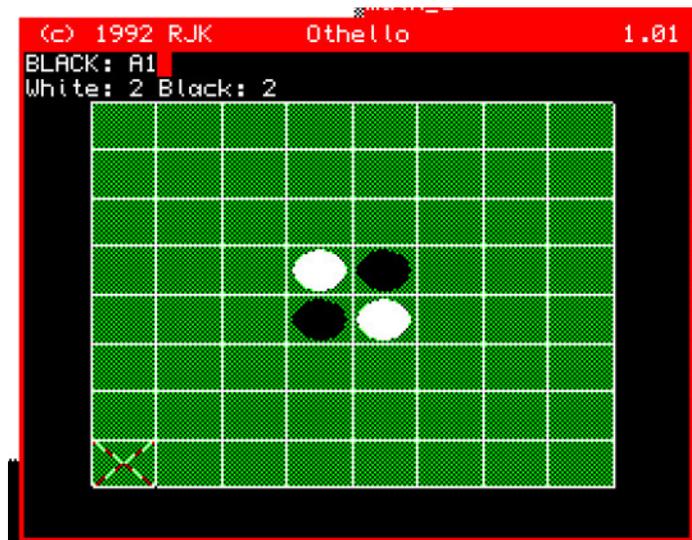
Richard Kettelwell's version

fairly weak game. This might be good for a beginning player.

Qthello.zip is written by Richard Greene. It will only run in TV mode so QDOS is needed. It has a very flat 3-D look to the game, but I think it looks better than some of the overhead view of other games. For QDOS this would be my favorite othello game.
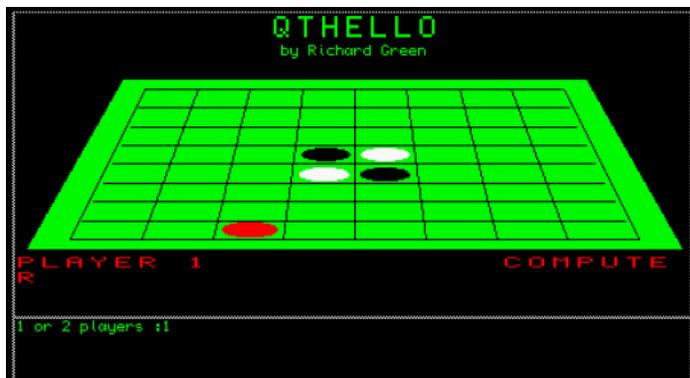
Richard Green's version

Reversi.zip is a Pointer Environment version of Othello by Frank Dibowski. As a Pointer Environment game it has the normal PE buttons for sleep and movement. All control is done via the mouse. It has a nice feature of flashing the moves after a move is made so that you can see what move was made and what other counters were flipped.

Despite needing TV mode, Othello by Richard Greene is probably my favorite. Next up would be the Frank D.'s version.
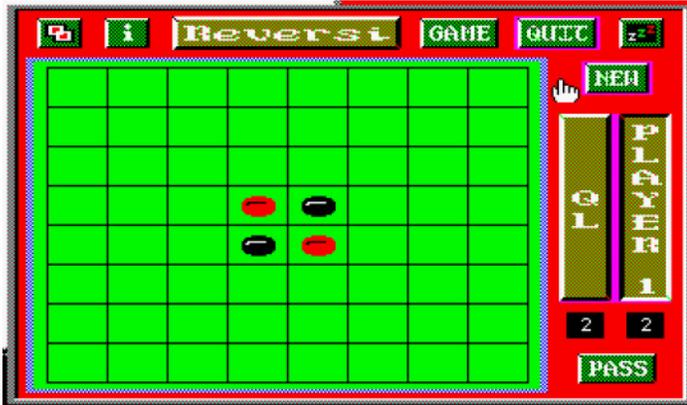
Frank Dibowski's version

New Release of UQLX

Recently there was a release for UQLX called uqlx2018. This fixed a few problems with UQLX for TCP networking. After this release, Martin Head found another issue, so out comes a new release called uqlx2018a.

The main fix is a minor one that caused problems for Martin and his assembly language written TCP driver.

There is also a new feature that will turn on the F1/F2 option when booting the QL.

The UQLX User Guide has been updated to reflect these fixes and the new feature.

The original uqlx2017 distribution is needed if one is installing from scratch and the new uqlx2018a binaries are then placed on top on the uqlx2017 distribution.

The new distribution is available from Dilwyn's web site.